Claims

1. A method for managing information exchanges among communicating objects in an
5 objects oriented client server system, said system including first and second object
oriented virtual machines running on counterpart first and second computers in
respective server and client roles, and a communication path connection between said
computers, said server virtual machine having a run-time environment, the method
comprising the steps of:

10

(a) generating a local object at the client machine operable as a proxy to a
remote object resident at the server machine;

(b) referencing the local object by an application executing at the client machine
15   and causing the local object to marshal parameters and send a process level call
request to the server machine;

(c) responsive to said request by the server machine's run time environment,
said run time environment causing the parameters in the request to become
20   unmarshaled, said remote object to be executed, the results of the execution
marshaled, and a process level return sent to the client machine; and

(d) responsive to said reply by the local object operable as a proxy, unmarshaling
the results from said reply.

25

2. The method according to claim 1, wherein said process call level requests and replies
are generated in an alternating manner.

30

3. The method according to claim 1, wherein the local object when operating as a proxy
at the client machine and the run-time environment when operating at the server
machine perform respectively as stubs.

5

10

15

20

25

30

4. A method for managing information exchanges between an application executing at a object oriented virtual machine operable as a client and a remote object resident at another object oriented virtual machine operable as a server, said server machine

5   having a run-time environment, said client and server having a communication path connection there-between, said communication path connection being operable under a process for originating and sending byte level messages therebetween, comprising the steps of:

10   (a) providing a local object resident at the client machine operable as a proxy
         stub to the remote object and providing a description of the remote object to
         enable said run-time environment to also operate as a stub;

         (b) responsive to a client application call to the local object, marshaling
15   parameters and causing a process level call request to be sent to the remote
         object at the server machine, said sending of the request further including
         mapping said process level call request into counterpart byte string level
         messages and transmitting said messages to the server machine;

20   (c) responsive to said request messages by the server machine's run-time
         environment, mapping said messages into a process level call request,
         unmarshaling the parameters, invoking and executing the remote object,
         marshaling the results, forming a process level reply, mapping said reply into
         string byte messages, and transmitting said reply messages to the client
25   machine; and

         (d) responsive to the reply messages by the proxy at the client machine, mapping
         said reply messages into a process level reply, and unmarshaling the results.

30

21

5. The method according to claim 4, wherein said object-oriented virtual machines include Java virtual machines, and further wherein the remote object is an applet, and the local object is an interface description.

5

6. An article of manufacture comprising a machine readable memory having stored therein a plurality of processor executable control program steps for managing information exchanges among communicating objects in an objects oriented client server system, said system including first and second object oriented virtual machines
10 running on counterpart first and second computers in respective server and client roles, and a communication path connection between said computers, said server virtual machine having a run-time environment, said control program steps include:

(a) a control program step for generating a local object at the client machine
15 operable as a proxy to a remote object resident at the server machine;

(b) a control program step for referencing the local object by an application executing at the client machine and causing the local object to marshal parameters and send a process level call request to the server machine;

20

(c) a control program step for responsive to said request by the server machine's run time environment, said run time environment causing the parameters in the request to become unmarshaled, said remote object to be executed, the results of the execution marshaled, and a process level return sent to the client machine;
25 and

(d) a control program step for responsive to said reply by the local object operable as a proxy, unmarshaling the results from said reply.

30

22